MetaRule

Multibyte Chars

Sean Barnum, Cigital, Inc. [vita[1]]

2007-03-29

Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 6254 bytes

Attack Category	• Malicious Input
Vulnerability Category	• Multibyte Character • Buffer Overflow
Software Context	• String Management • String Conversion MACROS
Location	
Description	Specifying number of bytes instead of number of characters for mutltibyte character operations can create serious reliability and security problems. Many of the multibyte character (MBC) functions are commonly misused by the programmer passing in "number of bytes" for the buffer size instead of "number of characters". In these situations, the routine thinks it is given a "250" character buffer (which is 500 bytes long) when in fact the user gave it only a 250 _byte_ buffer. Hence, the routine could easily overrun the buffer because the programmer erroneously specified the length. In particular, using the MultiByteToWideChar() function incorrectly can compromise the security of your application. Calling the MultiByteToWideChar function can easily cause a buffer overrun because the size of the Out buffer equals the number of WCHARs that Out string can hold, while the size of the In buffer equals the number of bytes. This can lure the programmer into erroneously specifying the number of bytes for both.
APIs	**Function Name** / **Comments** MultiByteToWideChar / winnls.h
Method of Attack	If the buffer size for a multibyte operation is specified incorrectly and an attacker can control the input to such a function, arbitrary code execution could result.
Exception Criteria	

[1] http://buildsecurityin.us-cert.gov/bsi-rules/35-BSI.html (Barnum, Sean)

<table>
<tr><td><b>Solutions</b></td><td>
<table>
<tr><th>Solution Applicability</th><th>Solution Description</th><th>Solution Efficacy</th></tr>
<tr><td>An multibyte character operation that requires the size of a buffer to be specified.</td><td>Ensure that the size of the output buffer equals or exceeds the number of WCHARs (plus the NULL character) that the output string can contain.
For an output buffer allocated locally (on the stack) a generic size calculation of "sizeof(buffer) / sizeof(buffer[0])" should give the correct result. If one uses "sizeof(buffer)" to specify the buffer size, this will be incorrect when the buffer is composed of wide characters.
For an output buffer allocated on the heap, a literal constant should be used to specify the buffer size. This constant should be the same constant used when specifying the number of characters to allocate the buffer.</td><td>Effective. But always check to verify that usage verifies what is expected by the particular API.</td></tr>
</table>
</td></tr>
<tr><td><b>Signature Details</b></td><td>int MultiByteToWideChar(</td></tr>
</table>

<table>
<tr><td></td><td>UINT CodePage, // code page
DWORD dwFlags, // character-type options
LPCSTR lpMultiByteStr, // string to map
int cbMultiByte, // number of bytes in string
LPWSTR lpWideCharStr, // wide-character buffer
int cchWideChar // size of buffer
);</td></tr>
<tr><td><b>Examples of Incorrect Code</b></td><td>char *UserName = "Jane Smith"; // ASCII user name
WCHAR wszUserName[UNLEN+1]; // Unicode user name

MultiByteToWideChar(CP_ACP, 0, UserName, strlen(UserName)+1, wszUserName, sizeof(wszUserName)); // Buffer size mis-specified!</td></tr>
<tr><td><b>Examples of Corrected Code</b></td><td>char *UserName = "Jane Smith"; // ASCII user name
WCHAR wszUserName[UNLEN+1]; // Unicode user name

MultiByteToWideChar(CP_ACP, 0, UserName, strlen(UserName)+1, wszUserName, sizeof(wszUserName)/ sizeof(wszUserName[0])); // Size correctly specified</td></tr>
<tr><td><b>Source References</b></td><td>• Rough Auditing Tool for Security (RATS)[2]
• Howard, Michael & LeBlanc, David C. <i>Writing Secure Code, 2nd ed</i>. Redmond, WA: Microsoft Press, 2002, ISBN: 0735617228.
• Microsoft Security Bulletin MS01-023. Unchecked Buffer in ISAPI Extension Could Enable Compromise of IIS 5.0 Server[3] (2003).</td></tr>
<tr><td><b>Recommended Resource</b></td><td>• MSDN MultiByteToWideChar reference[4]</td></tr>
<tr><td><b>Discriminant Set</b></td><td><table>
<tr><td><b>Operating System</b></td><td>• Windows</td></tr>
<tr><td><b>Language</b></td><td>• C
• C++</td></tr>
</table></td></tr>
</table>

Cigital, Inc. Copyright

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.

[1] mailto:copyright@cigital.com